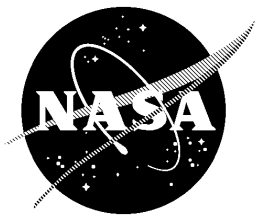
MISSION OPERATIONS AND DATA SYSTEMS DIRECTORATE

TDRSS Unscheduled Time and Nascom Information Distribution

NCC98 Issues 27 and 37
Version 1
(Review)

September 1996



National Aeronautics and SpaceAdministration

Goddard Space Flight Center
Greenbelt, Maryland

Abstract

TDRSS Unscheduled Time is provided to customers of the Space Network (SN) to indicate how customers may be able to obtain additional SN support. Information on SN customers' use of the IP Operational Network (IONET) will be provided to Nascom operations to aid in managing the IONET. This paper describes an investigation of methods to provide customers and Nascom operations access to this information using World Wide Web and Web/database technology. A literature survey identified many relevant products, with more being announced almost daily. Demonstrations of two similar systems developed for Goddard Spaceflight Center (GSFC) were viewed. A strawman architecture for the NCCDS was developed, with particular attention to security. The complete analysis of the security risks is provided in Appendix B (TBS); no major issues are foreseen. Because of the rapid development of this technology, further analysis and prototyping is recommended before a product selection is made.

Contents

APPENDIX B. SECURITY ANALYSIS

Section 1 . Introduction

1.1 Background

This paper presents an analysis of methods for providing the TDRSS Unscheduled Time (TUT) to Space Network (SN) customers and information on the scheduled data traffic between the SN ground terminals and SN customers to the Nascom Operations Center. This issue arose as a result of RIDs 50 and 51, written on the November 1995 NCC System Requirements Document. The original requirements specified a formatted message; the RIDs suggested an email message as well, and suggested that other approaches that conform to the client/server architecture may be more cost-effective for development and operation. NCC98 Issue Number 27, Action Item 27.2, specifies an analysis, including security risks, of the Oracle WebServer product as a means of providing TUT information to SN customers.

With the transition of the Nascom 4800 Bit Block network to the IP Operational Network (IONET) and the elimination of the Control and Status System (CSS) and its associated Nascom Event Schedule (NES) message, a similar issue arises: the Nascom operations center still needs information on the scheduled use of the Nascom network. NCC98 Issue Number 37, Action Item 37.1, specifies the same analysis for Nascom schedules as for TUT.

The Network Control Center Data System (NCCDS) System Requirements Document (SRD), Revision 1 (CCB Review), has been modified to require that the TUT data be accessible remotely for use by external systems and for displaying in a human-readable form, and that the Nascom schedule be accessible remotely for display in a human-readable form.

This paper addresses both issues for the human-readable form of the information. The investigation was broadened from simply examining the Oracle WebServer to looking in general at methods for providing interfaces to databases on internets and intranets. NCC98 Issue Number 27, Action Item 27.3, specifies a design of a distributed database access TUT service; this study does not address this Action Item at this time.

1.2 Definition of terms

TDRS Unscheduled Time is a table indicating unscheduled resources of the TDRSs of the Space Network. It consists of start and stop times of unscheduled use of each TDRS and Single Access (SA), Multiple Access Forward (MAF), and S-band Multiple Access Forward (SMAF) antennas, and Multiple Access Return (MAR) and S-band Multiple Access Return (SMAR) links.. This data is essentially the unused time in the schedule, with a few adjustments due to flexible events with flexible start and stop times and/or flexible resources. The TUT is not sensitive, and will be made available to all customers on the IONET.

The information to be provided to Nascom is essentially the contents of the baseline NES and the Nascom Event Cancel (NEC) message, without the MDM setup parameters (which will no longer be relevant in an IP network). Because this information identifies actual schedule

information for all SN users of IONET, it is assumed to be more sensitive than the TUT information and should be made available to the Nascom Operations Center only.

1.3 Overview of the paper

Section 2 of the paper describes the guidelines used in the study, a strawman architecture, and analysis of the details of the implementation. Section 3 presents conclusions from the analysis. Appendix A presents a summary of methods for implementing data presentation using the combination of World Wide Web and database technology. Appendix B (TBS) presents an analysis of the security risks associated with the approach.

1.4 References

The NCCDS requirements are contained in

Network Control Center Data System (NCCDS) System Requirements, 1998, Revision 1 (CCB Review), August 1996, 530-SRD-NCCDS/1998

The technology of database publishing on the Web is very new and products are appearing rapidly. Some useful references are:

Computer, IEEE Computer Society Press, Los Alamitos, California

Byte, McGraw-Hill Companies, Peterborough, New Hampshire, 1996 - the September 1996 issue has an evaluation of three new products

Curt Lang and Jeff Chow, *Database Publishing on the Web & Intranets,* The Coriolis Group, Inc., Scottsdale, Arizona 1996

Brian Jepson, *World Wide Web Database Programming for Windows NT*, John Wiley and Sons, New York, 1996

Section 2. Analysis

2.1 Guidelines

The goal was to evaluate methods of providing the TUT and Nascom schedule. The guidelines used in this analysis were:

a. Use current technology and paradigms, such as World Wide Web browsers for database access, multi-tier client/server architecture

b. Minimize custom software to be developed, both for the NCCDS and for the customers

c. Reduce security risks to acceptable levels. Limit external access to reading TUT and NES information only.

The paradigm of using the World Wide Web browsers and servers as interfaces to corporate databases on the Internet and intranets is becoming very popular:

- it provides the familiar Web user interface
- it provides access to the power of back-end database systems
- it is platform-independent, using the standard HTML Web interface
- it provides configuration control by insuring that the customer has the latest versions of pages because they are downloaded over the Web
- it requires no custom software development on the customer's system: any standard Web browser may be used.

For these reasons (and because the NCC98 Issue refers to the Oracle WebServer), we have concentrated on this approach. Appendix A describes the various architectures being used to implement this capability and mentions some of the current products available.

Other approaches that were considered in this investigation are:

a. FTP or email the data to the users
This is easy to do, but the user does not get the benefit of the Web UI. However, this could be provided as a supplement to the Web approach: if the customer wants the data sent directly to his/her system via FTP or email, the Web page could provide a button to do that.

b. Providing external customer read access to the SPSR database, for example, through SQL*Net
This approach requires extra development effort for the users, introduces some configuration management problems (if users have different platforms and operating systems), poses higher security risks (e.g., a SQL proxy for the firewall) and has no increased benefits for providing the human-readable TUT and Nascom schedules.

2.2 Strawman architecture

Based on the above decisions, the strawman architecture shown in Figure 2-1 was developed. This architecture meets the functional and security requirements; specific details are to be refined during prototyping and product selection activities.

The figure shows the logical flow of information. The customer, using a standard Web browser on the platform of his/her choice, accesses the URL for the TUT or Nascom information. The NCCDS Web server returns an HTML form to the user to specify search criteria such as time frame, TDRS, antenna, etc. The customer then submits the query to the NCCDS Web server, which passes it to a server extension to acquire the data requested and format it in a dynamic HTML page, which the Web server then returns to the customer.

The TUT information is generated periodically as specified by the NCC operator, or manually by the NCC operator (at the request of a customer). The information is written to a flat file and then sent to the Web server platform via FTP. This design minimizes the security risk, since the information flow is from the internal NCCDS through the firewall to the Web server. This assumes that the database extension server can execute queries against structured flat files; if not, a small local database on the Web server may be needed.

An alternative approach is shown for the Nascom information. The information is generated (e.g., by a stored procedure in the NCCDS database) whenever a SHO or SHO Cancel for an event is transmitted, stored in another table in the NCCDS database and read via an ODBC query from the Web server. Although this involves more complex communication through the firewall, it is sufficiently limited that the security risk is acceptable. Both approaches (ODBC or FTP) meet all the identified requirements.

The location of the Web server in the "DMZ" lets the firewall filter both outside connections to the Web server and the Web server connections to the SPSR database. Security can be increased in several ways. Oracle provides accounts with user IDs, passwords, privileges, roles and views; these all serve to limit access from the Web server. The Secure Socket Layer (SSL) protocol Version 2, which should be available in the near future for Web servers and clients, provides identification and authentication (I&A) of external users and encryption of data. Oracle's Secure Network Services (SNS) supports secure communications between the Oracle 7 server and the Oracle WebServer, even through a firewall; in this case the Web server must be configured as a bastion host.

For the TUT customers there is no requirement for I&A; however, as mentioned previously, the Nascom information is assumed to be more sensitive and an I&A function may be required. SSL/https are two ways of providing this security; Kerberos is another. NCSA is one source of Kerberized servers and browsers.

The "configuration management" mentioned previously is provided by the stored HTML pages on the Web server. These are sent to the customer, so that he/she always has the latest version. These pages can contain embedded code such as JavaScript to provide some validation at the client side.

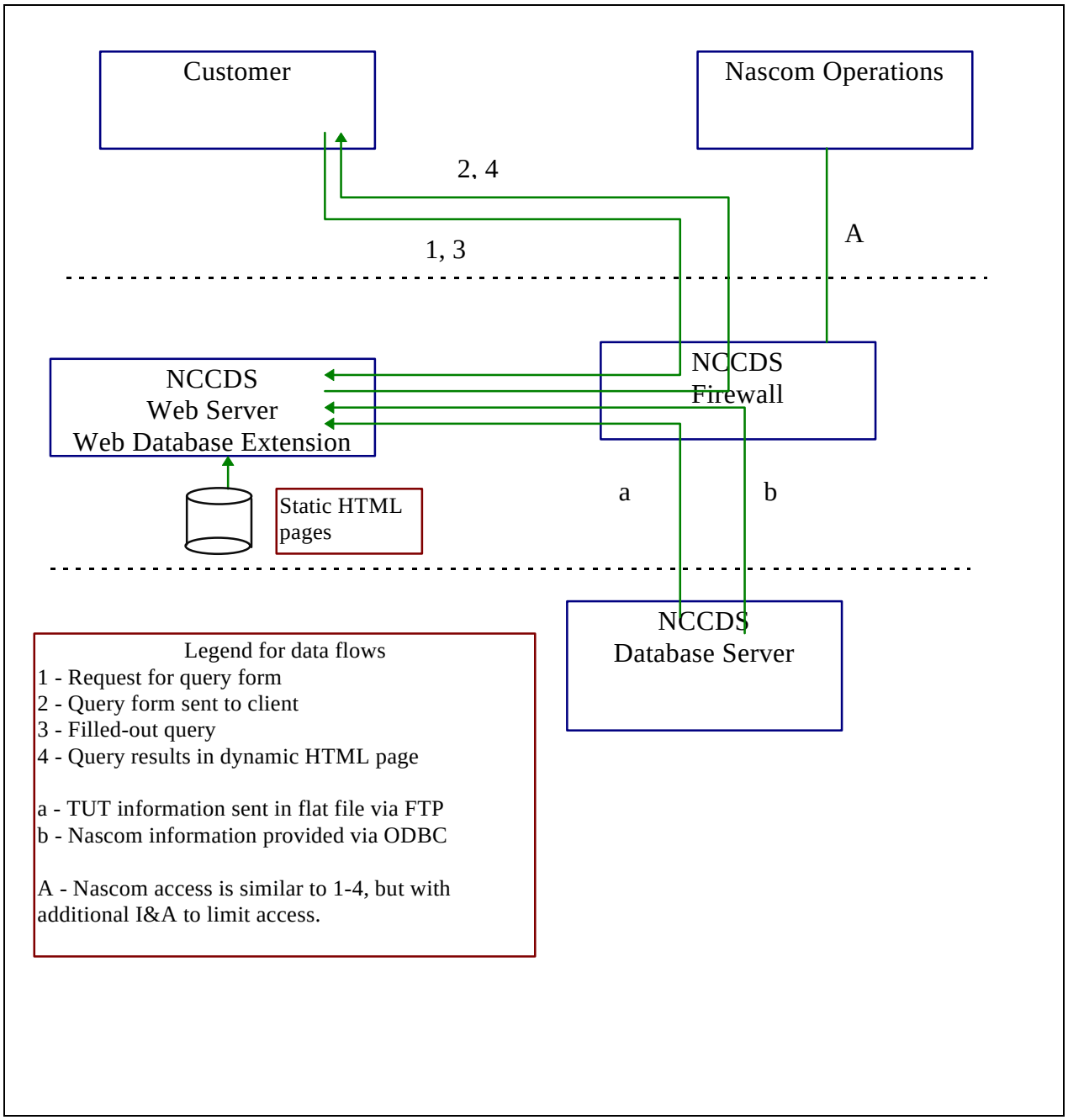


Figure 2-1. Strawman architecture for NCCDS Web Server for TUT and Nascom information

2.3 TUT computation

The TUT information is not represented directly in the NCCDS database, but must be generated from the data by some software process. Because the TUT data can encompass periods of the schedule in which flexible events may reside, the flexible events must in effect be "fixed" in order for a reasonable TUT data to be provided. This "fixing" may require

selecting a TDRS and SA antenna as well as start and stop times, and is sufficiently complex that it will probably be done by a subset of the software that generates the schedule.

A related question is when the data will be generated. Because it currently appears that it may require considerable time and processor resources to generate it, the requirement is that the NCC operator may specify that it be done periodically, but can also manually initiate it. If performance measurements on the delivered system indicate that TUT calculation is not a significant burden and can be done reasonably quickly, the requirement may be changed to have the TUT data generated on customer demand.

2.4 Nascom Schedule Computation

Because the Nascom information is derived from events that have been sent to the ground terminals and thus are fixed, the NCCDS database contains the information in a fairly direct manner. Analysis of the SPSR database design of August 1996 shows that 34 tables must be accessed to generate the information, and that the software could be done in PL/SQL (in a stored procedure) or in C/C++ with embedded SQL; the number of tables accessed may be reduced by minor database changes. The resulting information may be stored in another table in the NCCDS database or transferred to the Web server via database replication or simply sent by FTP.

An alternative approach is to implement the NES transmission to the Nascom CSS as originally planned but to send those messages to the Web server rather than to CSS. The Web server will collect this information and present it to the Nascom user via the Web. Because the SPSR will still have to generate and transmit the message to SDPF for events supported by that facility, and that message is nearly identical to the NES, much of the SDPF software could be reused. This approach can be extended to have the CCS send the Nascom Reconfiguration Requests (NRRs) to the Web server, so that Nascom operations can have access to the reconfigured data streams as well as to the original data streams.

One problem with providing only the composite Nascom schedule is that additions and deletions to the schedule is not readily apparent. Another problem is that the schedule shows the initial traffic configuration, but this changes in real time via reconfigurations requested by the customer. Further discussion with Nascom is needed to determine the complete Nascom operations requirements.

2.5 Platform and operating system

In comparison to most World Wide Web sites, the TUT Web server has a very small number of users and thus it is possible to use a low end workstation or a PC-based system to host the Web server. Although it is preferable to minimize the number of different hardware platforms and operating systems for ease in configuring and managing the NCCDS, there is a significant cost difference between a low-end UNIX workstation and a Pentium server.

A second issue is the operating system: Microsoft's Windows NT operating system is becoming the standard for many Web products. Although the additional complexity of managing a different operating system in the NCCDS must be considered, the use of standards-based COTS products for the Web server avoids the risk of being locked into

proprietary vendor software (note that Windows NT provides a POSIX interface). In addition, the increasing market share of Windows NT at the expense of UNIX is a factor that may influence future NCCDS development and maintenance.

Section 3. Products

There are many products available now for database publishing on the Web. Because of the rapid changes in this area (several major products were just released in the last three months), a listing of products is not provided. The different architectures that have been adopted are summarized in Appendix A, with a brief mention of some of the products that fall into each category. The Oracle WebServer is a viable candidate, but only one of many. As described above, some custom software must be developed because the information cannot be directly extracted from the NCCDS database with simple SQL queries.

3.1 Demonstration Systems

As part of the analysis of this work, two demonstration systems were viewed. An evaluation of PowerBuilder 5.0 is ongoing, and results from that study, when available, will be incorporated in this paper.

3.1.1 ISTP Project (Oracle WebServer)

The Oracle WebServer can connect with Oracle databases using the native Oracle interfaces. It includes security features: Basic and Digest Authentication, IP and domain-based restrictions, user logins, Secure Socket Layer (SSL) protocol, and Oracle Secure Network Services (SNS), which is compatible with many popular firewall products. It runs on Windows NT and HP-UX as well as other platforms. It supports CGI 1.1, and has an integrated Java runtime environment.

Rusty Whitman demonstrated a system he developed for ISTP to provide customers with information on available science data. This system uses Oracle WebServer Version 1 and was demonstrated with the Netscape Navigator Web browser. It can be used virtually out of the box to browse databases, select tables and columns and run queries against them. Stored procedures can be installed on the database server to process the data before viewing. It uses PL/SQL as its database agent, allowing very complex programs to be developed if necessary.

3.1.2 Data Distribution System (DDS) (Netscape Web Server with CGI Extensions)

George Kallarakal demonstrated the Web portion of the Data Distribution System (DDS) developed for the GSFC Data Distribution Facility. It provides a browsable index of available science data using Netscape Web server with extensions written in C and embedded SQL. These programs query the Oracle database and build dynamic HTML pages showing the available data products based on the input query. In addition, the system can send email to customers when data is available and can send the customers the data through FTP or they can download it through the WWW.

This is a conventional approach using CGI extensions and a third-generation language; however, the code involved is fairly short (about 150-200 lines to formulate a query and return the HTML-formatted page).

3.2 PowerBuilder 5.0

This newest release of PowerBuilder provides the capability of converting an application to a Netscape Plug-in that can then be downloaded over the network and executed on the client machine. The value of this approach is that database applications can be built rapidly, and the downloaded application provides configuration management of the distributed applications. Dave Messent is currently involved in an evaluation of this product.

Section 4. Conclusions

It is clear from analysis and demonstration systems that it is technically feasible to use a World Wide Web interface to deliver the TUT and Nascom information. Although none of the demonstration systems were concerned with security to the extent the NCCDS is, there are a number of approaches to insuring the security of the NCCDS using this technology and security is not viewed as a major risk.

The functionality needed is available now in the commercial products, but the security aspects (such as running through a firewall, I&A, etc.) are only now being addressed by the vendors. Because of the rapid developments in this area of Web database publishing we recommend that further analysis and prototyping of some products be performed before a product selection is made.

Appendix A. Database Web Server Architectures

A.1 Server-side Extensions

A.1.1 Common Gateway Interface (GCI)

This is provided by all modern web servers. They provide this interface to programs that are executed in response to some input. The parameters to the program are usually passed in environment variables. One advantage to this method is that it is probably the most portable method of adding extensions. A disadvantage is that each program is started as a separate process, an action that is time-consuming and may lead to performance problems on heavily-used servers. The programming language is typically C/C++, Perl or shell scripts; Java is now being used for this programs as well.

A.1.2 Hybrid CGI

This is a simple modification that overcomes the performance problem of straight CGI. Most of the work of the server extension is performed by a separate process that is always executing (e.g., a UNIX daemon or a Windows NT service). The CGI program is a small interface that passes requests to the service and passes the results back to the Web server. This does not require any additional support from the Server, beyond the usual CGI.

One product that uses this architecture is Cold Fusion from Allaire. It has a small CGI that passes Web pages to a Windows NT service that connects to databases using ODBC (see below). It uses a small set of database access commands that are embedded in the HTML pages.

A.1.3 Web Server Application Program Interface (API)

A Web API allows compiled server extensions to link directly to the Web server (e.g., Dynamic Link Library in Windows NT). This method provides the fastest performance, and may enable the extensions to provide more functionality than a CGI program. The risk is that the extension essentially becomes part of the Web server, and any software error in the extension can bring down the whole server. In contrast, errors in a CGI program only cause loss of one query.

Both Netscape and Microsoft provide an API for their Web server. Microsoft Internet Database Connector uses MSAPI to interface with the Microsoft Internet Information Server.

A.2 Client-side extensions

These are extensions to the Web browser running on the user's machine. These typically provide improved field validation, but the newer extensions can provide functionality outside of the browser.

A.2.1 Helper applications

These are stand-alone applications that cooperate in some degree with the browser. An example is the Adobe Acrobat Reader, to read PDF files downloaded by a browser.

A.2.2 Plug-ins

These are similar to helpers, but they are more closely coupled to the browser. The user must download and install them. The PowerBuilder 5.0 Web Plug-ins are an example.

A.2.3 Java applets

These are small programs written in Java that are downloaded with the HTML page and executed by the browser during the display of the page. An advantage of applets is that they are downloaded when needed, thus insuring that the user has the latest version.

A product using this approach is JAGG from Bullet Proof Corporation. It includes jagg.class that is compiled in a Java applet and executed on the client side, and jagg.exe that interfaces with the Web server via CGI and the database via ODBC.

A.2.4 Scripts

Scripts are embedded in an HTML page. They are useful for providing input validation. JavaScript and VBScript are the two common scripts available today. Netscape's Livewire Pro uses JavaScript to provide database access.

A.3 Mixed Database Systems

This is a hybrid of the traditional database world and Web world. The Web is used to download an application that can then access the database using the traditional client/server configuration. The new PowerBuilder 5.0 provides such a capability. This approach may be useful for applications where the data is to be further processed in the user's computer (e.g., Nascom information): the Web provides the configuration management of the tool.

A.4 Database Connectivity

There are several possibilities for the interface between the server extension and the database. One possibility is to use the native database protocols. For example, a CGI program in C with embedded SQL could be used with Oracle 7 to provide the database access (the DDP demonstration referenced in Section 3 is an example).

The Oracle WebServer is similar: the database access extension is well integrated with the Web server, and supports native Oracle SQL.

A more portable approach (although one with possibly slower performance) is to use the Open Database Connectivity (ODBC) API, which is available for many different database systems. It provides an interface between the server extension and the ODBC database drivers that are specific for each database product.

Appendix B. Security Analysis

Two security levels are anticipated for access to the Web server. The TUT is not sensitive and will be made available to all NCC customers. Because NES is sensitive and limited to one customer a need for confidentiality as well as authentication is assumed. Access control needs to occur in two data paths: between the external customer and the Web server and between the Web server and the SPSR. The two paths are addressed separately.

It is anticipated that the current firewall architecture will not be affected materially by the Web server. The firewall architecture assumed a third interface would be provided to create a “DMZ” where Web servers and other semi-secure "Internet" servers would be located.

B.1 Authentication to the Web Server

Several access control points and levels are available including:

- Authentication at Session Start at the Firewall.
- Authentication and Encryption at the Firewall.
- Authentication at Session Start at the Web Server.
- Authentication and Encryption at the Web Server.
- Authentication before the firewall

Security export restrictions may apply to future NCC customers if strong encryption is selected. However, this is a minor concern since weakened versions of most protocols are available.

B.1.1 Authentication at Session Start at the Firewall.

Authentication at the start of a session at the firewall uses reusable passwords, one-time password (OTP) schemes such as S/Key or proprietary systems such as SecureID cards. Reusable passwords can be sniffed or key strokes monitored and are considered very weak security. The OTP systems are preferable to reusable passwords but only provide better authentication at the session start. After the initial login the session is still subject to being monitored or highjacked. Most firewalls only support two or three authentication systems,

thus limiting purchase options to a few combinations. Simple password authentication with unencrypted data streams is not recommended. If NES data has a very low security requirement then at least an OTP system should be used. S/Key is a minimal system readily available in the public domain.

Protocols that authenticate each packet are also capable of encrypting each packet. They are considered in the Authentication and Encryption section below.

B.1.2 Authentication and Encryption at the Firewall.

Most firewall vendors offer products that provide authentication with encryption at the firewall. Each system is limited to working only with the vendors own firewall and most have the client available only on Windows based PC's.

SmartGate from V-One is a standalone product that can work with a number of firewalls from different vendors and provides several simple authentication options as well as support for encryption based on FORTEZZA and V-One's proprietary SmartCat system. In addition, SmartGate has an HTTP proxy that can be set to limit access by particular users to specific files or URL's, allowing security administration for the Web server to be done at the authentication server rather than, or in addition to, at the server itself. SmartGate also supports a key distribution and activation system to help get new users on-line faster and with less administrative work.

SmartGate should be considered further as a viable security option if NCC expects the use of the Web server by secure clients to grow or if many new protocols and systems are expected to be added to the NCC services. It is difficult to justify for one or just a few customers with limited access needs.

B.1.3 Authentication at Session Start at the Web Server.

Authentication only at the start of a session at the Web server uses essentially the same technology as authentication at the start of a session at the firewall. In addition to the same problems found in firewall authentication, CGI's would need to be written to access the OTP servers when protected areas are accessed. Although possible, this option is not practical.

B.1.4 Authentication and Encryption at the Web Server.

There are several systems available and in development for session authentication and encryption at the Web server. The most promising are the Secure Sockets Layer (SSL)

protocol, Secure HTTP (SHTTP) and Pretty Good Privacy (PGP). Kerberos may also be a viable option for the NCC.

B.1.4.1 SSL

SSL relies on third party Certificates to act as letters of introduction. The protocol was first aimed at authenticating the server to the client to assure the client of secure transactions. After the server is authenticated it may request authentication of the client.

In order for either client or server to be authenticated they must register with a Certificate Authority (CA). The CA issues a digitally signed certificate that essentially identifies the system presenting the certificate. The certificate is presented during session negotiation along with other parameters for establishing a session encrypted or authenticated with read and write session keys.

Because SSL is located on top of TCP/IP but below the application layer in the ISO stack it allows most system components to run unmodified or with minimal modification. Version 2.0 is well supported and readily available from several server vendors. Version 3.0 was recently released as a standards track candidate and is available from Netscape.

Since the certificate is issued only once and remains permanently resident on the certified server or client it may be difficult to check for certificate revocation or early expiration. There are only a few public Certification Authorities available at this time and there is a small fee for obtaining a certificate. Netscape has announced their Certification Server product. This would allow NCC to act as its own Certification Authority but, depending on cost, it may be hard to justify for a single secure customer.

SSL using 40 bit keys is generally considered weak security. However, a new "US" version is now emerging (no server support yet) that supports 128 bit keys and triple DES.

B.1.4.2 PGP

PGP is a popular Public Key signature system. Users have a Public Key and a Private Key. The Private Key can encrypt and/or sign messages which can be decrypted with the Public Key and it can decrypt messages encrypted with the Public Key. The Public Key cannot be used to decrypt messages encrypted using the Public Key. Confidentiality is provided for messages sent from the Public Key holder to the Private Key holder since only the Private

Key owner can open messages encrypted with his Public Key. Only the Private Key holder could have written a message that can be decrypted by the Public Key. If the Public Key is kept secret the message is confidential. If the Public Key is shared with others any one of them may decrypt and read the message.

The NCC could use a Private PGP Key to encrypt NES data on the SPSR server and transfer the encrypted file to the Web server. NCC customers who were provided with NCC's Public Key could download and interpret the data. Even the requests for download could be encrypted with the Public Key.

In the above scenario multiple NCC customers could see the same data by applying NCC's Public Key to broadcast data. For greater confidentiality both the NCC and each customer would have a key pair. The NCC would receive messages encoded with its Public Key and would transmit messages encoded with the individual customers Public Key. Since in each case only the Private Key holder could interpret the data confidentiality is assured.

The use of PGP would require the NES data be encrypted separately for each customer. A small amount of code would also be required for key management and program control to select the right key from the "key ring" if multiple customers are supported. Key management and program control are not significant problems with only one NES customer but the solution does not scale if other applications wish to use the Web server in a similar manner and the number of keys gets large.

NCC's private key would be stored in a file so if the Web server were to be compromised the key would be vulnerable to discovery. Other key storage locations would present operational complexity that should be avoided.

B.1.4.3 Kerberos

Because the NCC already has a Kerberos KDC in its architecture, using Kerberos to authenticate users on the Web server is sensible. The main concern is that few products support it directly at this time: this study has identified only the NCSA Web server and Mosaic browser..

Applications written using Kerberos calls and applications using the General Security Service (GSS)-API are expected to be incompatible. There is a high probability that the NCSA Web server and browser operate based on the direct Kerberos V4 calls instead of using the GSS-API. It is also likely that the available Kerberized browser and Web server version will not

support cross-realm authentication. Further analysis is needed to determine which interface to use in Web Kerberos applications. The NCC may end up with Web applications using Kerberos V4 and SPSR applications using Kerberos V5.

B.1.4.4 SHTTP

SHTTP is an application level security protocol. It primarily provides an interface to lower level security systems such as DES encryption or Kerberos. Because SHTTP is at the application level it provides the NCC with only marginal improvement over using Kerberos and the GSS-API. The SHTTP advantage is that several firewalls provide an SHTTP proxy and the protocol is supported by several Web servers and browsers. It could use the NCC Kerberos KDC for the authentication and encryption processing or more directly a DES or other encryption/authentication library. Since the application, server and browser would be interacting with SHTTP we may be able to supply Kerberos V5 ("NCC compliant") libraries for the SHTTP security calls. This possibility would require additional research but is attractive since it lets NCC use its existing Kerberos authentication system and provides more choices for Web servers and browsers.

B.1.4.5 FORTEZZA

Recently Netscape announced support for the FORTEZZA card by its Enterprise Server. FORTEZZA is the United States Government's security standard for inter-

and intra-federal communications. Private keys and authentication certificates are stored

on PCMCIA cards and protected through the use of personal identification

numbers (PINs). The card also contains the cryptographic engine. The user must "have" something and "know" something to access a system. A FORTEZZA card can store keys to multiple systems so there is an economy of scale if a user must access several government systems.

FORTEZZA is still an emerging technology and still not widely available though most firewall vendors have announced plans to support FORTEZZA access control. It's likely that the same card may be used to access additional future functions within the NCC. However, the FORTEZZA system includes a library with many API's so application programming for it is non-trivial.

Since the NCC anticipates only one initially secure client it is difficult to justify the expense of a card administration system. If some other NASA agency were to issue the FORTEZZA card to NCC's customer(s) this could be an attractive Web access solution.

B.2 Communicating between the Web server and internal Databases

The two most likely methods for transferring data between the SPSR to the Web server are to use FTP or to have the Web server act as an SQL client to the SPSR. Other methods may be available, such as e-mail, but are less secure, elegant or efficient.

B.2.1 FTP

From a firewall and security administration view point this is the preferred method of moving data from the SPSR to the Web server. Standard firewall configurations permit FTP sessions to originate from an internal system and to connect to an external system. The data could be transmitted in the clear since it is essentially traveling on the NCC OpsLAN. It could also be authenticated or encrypted using the Kerberized FTP provided with the NCC Kerberos system. In addition, some firewall proxies actually control the direction of file transfers (Put vs. Get), adding another security control option. Since the transfer is originating from the SPSR and going only one way, an attacker who compromised the Web server would have to find some other mechanism to attack the internal OpsLAN.

This solution solves the immediate problem for TUT, NES and other data that is updated periodically. It does not scale very well to interactive queries. However, it is the recommended solution for the current system since it is secure and can be implemented with minimal effort.

B.2.2 SQL

SQL is an elegant and conceptually attractive solution to data sharing. The solution would also support development of future SQL applications, possibly including interactive applications. SQL proxies are starting to emerge on many firewall platforms, encrypted and DCE-based (Kerberized) clients are also emerging and DES encrypted SQL links are available. Database "Views" can be used to control database access at the database itself.

However, the use of SQL to query the SPSR would require very good security control at the SPSR database. The first generation of SQL proxies is still fairly primitive, checking only the source IP address, port, destination database name and destination host. Encrypting the SQL link or using Kerberos does not add a great deal to NCC security since the data is only traversing the NCC OpsLAN. Since the Web server would be accessible to unauthorized users, a breakdown in the Web server could provide a path to the SPSR for SQL queries by unauthorized users.

Restricting SQL queries from the Web server to only trigger a predefined process might add a degree of security but complicates SPSR system design slightly. Another possibility is to disallow SQL queries from the Web server to the SPSR and simply use the SQL mechanism to periodically update a database or process on the Web server. It makes SQL analogous to using FTP but provides a basis for further development of SQL applications. As better tools become available and database expertise increases many security concerns can be expected to be addressed. The main danger is that future developers may enable two way queries/updates or eliminate or "streamline" predefined processes without due consideration of the security issues.

Note that much of the security focus is restricting access from external sites to the SPSR and other internal nodes. If the focus of security is moved to the SPSR and it is "hardened" sufficiently then many of the concerns are reduced significantly and SQL access to the SPSR becomes very attractive.

B.3 Conclusion

For periodic data transfers from the SPSR to the Web server, FTP enabled in only one direction is the preferred mechanism for a quick solution. SQL should continue to be watched and evaluated and may be able to replace FTP.

SHTTP should be investigated further to see if it may be implemented with NCC's Kerberos system as the underlying security mechanism.

Fall back alternatives to SHTTP would be SSL and PGP. To use SSL an accessible and reliable Certificate Authority (CA) would need to be found or obtained. Currently known CA's include the U.S. Post Office and VeriSign, but both must be reached via the Internet. If an acceptable CA is readily reachable or a certificate-generating system can be purchased at reasonable cost the SSL mechanism may be preferable to SHTTP due flexibility and support from many vendors.

Finally, basic password checking at the firewall and/or server or simple S/Key checking at the firewall may be sufficient during most of the application development phase. It is very likely that by the time the Web server goes on-line several improvements will have been made to Web server access control and new products are likely.